COUPLED MCMC WITH A RANDOMIZED ACCEPTANCE PROBABILITY

GEOFF K. NICHOLLS[†], COLIN FOX[*] AND ALEXIS MUIR WATT[†]

Abstract. We consider Metropolis Hastings MCMC with target $\pi(\theta)$ in cases where the log of the ratio of target distributions $D = \log(\pi(\theta')/\pi(\theta))$ is replaced by an estimator $\hat{D}(W)$. The estimator is based on m samples $W = (W_1, W_2, ..., W_m)$ from an independent online Monte Carlo simulation. Under some conditions on the distribution of $\hat{D}(W)$ the process resembles Metropolis Hastings MCMC with a randomized transition kernel. When this is the case there is a correction to the estimated acceptance probability which ensures that the target distribution remains the equilibrium distribution. The simplest versions of the *penalty method* of Ceperley and Dewing 1999 [6], the *universal algorithm* of Ball et al. 2003 [3] and the *single variable exchange algorithm* of Murray et al. 2006 [15] are special cases. In many applications of interest the correction terms cannot be computed. We consider approximate versions of the algorithms. We show that on average $O(m)$ of the samples realized by a simulation approximating a randomized chain of length n are exactly the same as those of a coupled (exact) randomized chain. Approximation biases Monte Carlo estimates with terms $O(1/m)$ or smaller. This should be compared to the Monte Carlo error which is $O(1/\sqrt{n})$.

1. Introduction

Monte Carlo simulation offers a direct route to statistical inference for many otherwise awkward fitting problems. The class of problems which may be treated using Markov chain Monte Carlo (MCMC) and Sequential Monte Carlo has grown a great deal since the core algorithms were proposed [12, 7]. One of the most important recent advances [11, 4, 2, 1] has given us pseudo-marginal MCMC algorithms which are useful for some doubly intractable distributions. Such distributions are hard to simulate as we cannot readily compute the ratio of densities at two values of the target variable. Algorithms with a pseudo-marginal target distribution put an estimate for the target distribution in the Monte Carlo state along with the target variable. In this paper we give a class of MCMC algorithms characterized by a pseudo-marginal transition probability kernel. In these algorithms the Monte Carlo state is just the target variable. The class as a whole seems to be unknown in statistical inference, though specific examples have appeared in various contexts in the physics literature.

The MCMC algorithms we describe are, for the most part, exact only in the case that the ratio estimator has a known log-normal distribution. By 'exact' we mean ergodic, with equilibrium distribution equal the desired target, not 'perfect' in the sense of Propp and Wilson 1996 [18] and 1998 [17]. The restriction to log-normal estimators corresponds to the case where we have a normal estimator for the log-likelihood. We know of no real problem which presents this feature. However, a ratio estimator may be approximately log-normal when there is an appropriate CLT. We show that an inexact MCMC algorithm may be adjusted to yield, with high probability, just the same samples as a suitably coupled exact algorithm, in any simulation of fixed length. At any fixed precision of the overall Monte Carlo estimate the approximation error may be zero.

The paper has five sections. In Section 2 we give a class of Metropolis Hastings MCMC algorithms with a pseudo-marginal transition kernel. This is randomized MCMC. In Section 3 we show that three existing MCMC algorithms for doubly intractable problems are special cases of the algorithm described in Section 2. In Section 4 we give a coupling-separation algorithm motivating the use of approximate MCMC. In Section 5 we give two very simple examples, chosen so that the coupling separation algorithm can be implemented exactly. We conclude with a brief discussion of the results in this paper.

2. Randomized Metropolis Hastings algorithms

2.1. **Standard MCMC.** We begin with the standard MCMC algorithm of Hastings 1970 [8] and Metropolis et al. 1955 [12]. We call this the s-algorithm (standard algorithm). Our notation follows Tierney [19].

Let $\theta \sim \pi(d\theta)$ be a target variable with state space E, having a distribution $\pi(d\theta) = \pi(\theta)\mu(d\theta)$ which has a density $\pi(\theta)$ with respect to a measure $\mu(d\theta)$ defined for sets in a sigma-algebra $\mathcal{E}$ of subsets of E. Let $Q(\theta, d\theta') = q(\theta, \theta')\mu(d\theta')$ be a Hastings proposal distribution with density q with respect to μ, satisfying $q(\theta, \theta') > 0 \Leftrightarrow q(\theta', \theta) > 0$. Let

$$h(\theta, \theta') = \frac{\pi(\theta')q(\theta', \theta)}{\pi(\theta)q(\theta, \theta')}$$

so that

$$\alpha(\theta, \theta') = \min\{1, h(\theta, \theta')\}$$

is the standard Metropolis Hastings acceptance probability. Let

$$p(\theta,\theta') = \begin{cases} q(\theta,\theta')\alpha(\theta,\theta') & \theta \neq \theta' \\ 0 & \theta = \theta' \end{cases} \tag{2.1}$$

be the zeroed transition probability density in the s-algorithm. Let

$$r(\theta) = 1 - \int_E p(\theta,\theta')\mu(d\theta')$$

give the probability for a rejection, and let $\mathbb{I}_{\theta=\theta'}$ be the indicator function for the event $\theta = \theta'$. The transition probability distribution for s-MCMC is

$$P(\theta, d\theta') = p(\theta,\theta')\mu(d\theta') + r(\theta)\mathbb{I}_{\theta=\theta'}.$$

2.2. **Randomized MCMC.** We now give algorithms with a randomized acceptance probability. We call these algorithms r-algorithms.

Let X be a real scalar random variable with probability density $\xi(x;\theta,\theta')$. We assume the support W of ξ is independent of θ and θ'. Let $f : W \to W$ be an *involution*, i.e. f satisfies $f(f(x)) = x$. The involution f can be thought of as pairing points $(x, f(x))$ in W. We assume that f has a derivative at ξ-a.e. $x \in W$. Examples of suitable involutions are the trivial involution $f(x) = x$, and the family of functions

$$f(x) = \frac{ax+b}{cx-a}$$

with $a^2 + bc \neq 0$.

Let

$$h_\xi(\theta,\theta';x) = h(\theta,\theta')\frac{\xi(f(x);\theta',\theta)}{\xi(x;\theta,\theta')}|f'(x)|$$

so that

$$\alpha_\xi(\theta,\theta';x) = \min\{1, h_\xi(\theta,\theta';x)\} \tag{2.2}$$

gives an acceptance probability which is randomized by X.

The r-algorithm is as follows.

Algorithm 1 (r-MCMC). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta,\cdot)$ and $x_t \sim \xi(\cdot;\theta,\theta')$.
(2) With probability $\alpha_\xi(\theta,\theta';x_t)$ set $\Theta_{t+1} = \theta'$ and otherwise set $\Theta_{t+1} = \theta$.

We now show that the r-algorithm simulates a transition kernel that is in detailed balance with π, so that $\{\Theta_t\}_{t=0,1,2,\ldots}$ is a Markov chain targeting π. The acceptance probability in the r-algorithm is

$$\alpha_\xi(\theta,\theta') = \int_W \xi(x;\theta,\theta')\alpha_\xi(\theta,\theta';x)dx \tag{2.3}$$

and we wish to establish detailed balance, i.e.,

$$\pi(\theta)q(\theta,\theta')\alpha_\xi(\theta,\theta') = \pi(\theta')q(\theta',\theta)\alpha_\xi(\theta',\theta). \tag{2.4}$$

Multiplying both sides of Eqn 2.2 by $\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta')$ shows that

$$\begin{aligned}&\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta')\alpha_\xi(\theta,\theta';x) = \\ &\min\left\{\pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta'), \pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)|f'(x)|\right\}.\end{aligned} \tag{2.5}$$

Similarly,

$$\begin{aligned}&\pi(\theta')q(\theta',\theta)\xi(y;\theta',\theta)\alpha_\xi(\theta',\theta;y) = \\ &\min\left\{\pi(\theta')q(\theta',\theta)\xi(y;\theta',\theta), \pi(\theta)q(\theta,\theta')\xi(f(y);\theta,\theta')|f'(y)|\right\}.\end{aligned} \tag{2.6}$$

In Eqn 2.6 set $y = f(x)$ so $x = f(y)$ and $f'(y) = 1/f'(x)$ by the inverse function theorem. It follows that

$$\begin{aligned}&\pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)\alpha_\xi(\theta',\theta;f(x))|f'(x)| = \\ &\min\left\{\pi(\theta')q(\theta',\theta)\xi(f(x);\theta',\theta)|f'(x)|, \pi(\theta)q(\theta,\theta')\xi(x;\theta,\theta')\right\}\end{aligned} \tag{2.7}$$

and this equation has RHS equal to the RHS of Eqn 2.5. It follows that the LHS of Eqns 2.5 and 2.7 are equal, i.e.,

$$\begin{aligned}&\pi(\theta)q(\theta,\theta')\alpha_\xi(\theta,\theta';x)\xi(x;\theta,\theta') = \\ &\qquad\qquad \pi(\theta')q(\theta',\theta)\alpha_\xi(\theta',\theta;f(x))\xi(f(x);\theta',\theta)|f'(x)|.\end{aligned} \tag{2.8}$$

This is 'very detailed balance'. Integrating Eqn 2.8 over all x in W gives detailed balance on average, Eqn 2.4. Besag et al 1995 [5] show that continuously indexed proposal distributions, each satisfying detailed balance on its own, can be used in MCMC. In our setup the transition kernels satisfy detailed balance in pairs, the kernel at x with the kernel at $f(x)$.

In the next section we mention algorithms with more than one randomization. We claim without proof that the properties of r-MCMC established in this and the next sub-section hold under the generalization from scalar to multivariate X. That is, if $X = (X_1, ..., X_K)$ has multivariate density $\xi(x;\theta,\theta')$ with support W in $\Re^K$, and f is an involution of W having Jacobian $f'(x)$ with determinant $|f'(x)|$, then the r-algorithm in Alg 1 targets π.

2.3. Properties of r-MCMC.

In this section and Appendices A and B we show that r-MCMC is less statistically efficient than the s-MCMC from which it is derived, and give a sufficient condition for the r-chain to inherit π-irreducibility and minorization from the corresponding s-chain. These results can be used to establish ergodicity in particular cases.

First, r-MCMC is less statistically efficient than the s-MCMC. We show (in Appendix A) that, for all θ, $\theta' \in E$,

$$\alpha_\xi(\theta, \theta') \leq \alpha(\theta, \theta'). \tag{2.9}$$

Since the two chains have the same proposal distribution $Q(\theta, d\theta')$, this puts the r-chain below the s-chain in the ordering of Peskun (1973) [16], and so r-chain estimators have greater asymptotic variance than corresponding s-chain estimators.

We next give a sufficient condition for r-chain π-irreducibility and minorization. Let $p_\xi(\theta, \theta')$ be given by Eqn 2.1, with $\alpha(\theta, \theta')$ replaced by $\alpha_\xi(\theta, \theta')$ (see Appendix B for details). Let $r_\xi(\theta) = 1 - \int_E p_\xi(\theta, \theta')\mu(d\theta')$ and

$$P_\xi(\theta, d\theta') = p_\xi(\theta, \theta')\mu(d\theta') + r_\xi(\theta)\mathbb{I}_{\theta=\theta'}.$$

Suppose there exists a constant $\epsilon > 0$ such that

$$\Pr\left(\frac{\xi(f(X); \theta', \theta)}{\xi(X; \theta, \theta')}|f'(X)| \geq 1\right) \geq \epsilon, \tag{2.10}$$

for $X \sim \xi(\cdot; \theta, \theta')$, independent of θ and θ' in E. If ξ does not depend on θ, θ', as in the penalty method example [6] below, condition Eqn 2.10 is automatically satisfied (see Appendix B).

We show (in Appendix B) that if Eqn 2.10 holds then

$$P_\xi(\theta, C) \geq \epsilon P(\theta, C). \tag{2.11}$$

Because the s-chain is π-irreducible by assumption, for each $C \in \mathcal{E}$ such that $\pi(C) > 0$ and for π-a.e. $\theta \in E$ there exists $n = n(\theta, A)$ such that $P^{(n)}(\theta, C) > 0$. However, if Eqn 2.11 holds then $P_\xi^{(n)}(\theta, C) \geq \epsilon^n P^{(n)}(\theta, C)$, so the r-chain is π-irreducible also.

Suppose the s-chain satisfies a minorization condition $M(m, \beta, C, \nu)$. This means that there exist $m \geq 1$, $\beta > 0$, a set $C \in \mathcal{E}$ and a probability measure ν on $\mathcal{E}$ such that $\nu(C) > 0$ and $P^{(m)}(x, B) \geq \beta\nu(B)$ for all $x \in C$ and all $B \in \mathcal{E}$. It follows from Eqn 2.11 that the r-chain satisfies a minorization condition $M(m, \epsilon^m\beta, C, \nu)$. If the s-chain is uniformly ergodic, then so is the r-chain.

2.4. **Example.** We now give a simple example illustrating the r-algorithm. Suppose we have an s-algorithm targeting $\pi(\theta)$ with symmetric proposal $q(\theta, \theta') = q(\theta', \theta)$ and acceptance probability

$$\alpha(\theta, \theta') = \min\left\{1, \frac{\pi(\theta')}{\pi(\theta)}\right\}.$$

We can randomize this acceptance with a normal density $\xi(x; \theta, \theta') = N(x; a, b)$ having mean $a = \log(\pi(\theta')/\pi(\theta))$ and variance $b = 1$, and use the identity involution $f(x) = x$, to get the following r-algorithm:

Algorithm 2 (Example). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta, \cdot)$ and $z_t \sim N(0,1)$. Set

$$x_t = \log(\pi(\theta')/\pi(\theta)) + z_t.$$

(2) With probability

$$\alpha_\xi(\theta, \theta'; x_t) = \min\left\{1, \left(\frac{\pi(\theta')}{\pi(\theta)}\right)^{1-2x_t}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

The example algorithm satisfies detailed balance with respect to π. The correction term can be interpreted as kind of random tempering, since the random power flattens the target distribution when $0 \leq 1 - 2x_t < 1$. However this 'tempering' does not decrease the integrated autocorrelation time of the Markov chain. As shown above, the r-chain is dominated by the s-chain in the Peskun ordering, and hence the integrated autocorrelation time is not decreased.

3. Known randomized Metropolis Hastings algorithms

We now describe some existing MCMC algorithms for doubly intractable distributions and show that they are r-algorithms. From this point on, we assume for ease of exposition that the proposal density is symmetric, $q(\theta, \theta') = q(\theta', \theta)$, since the Hastings extensions are straightforward. Suppose

$$D(\theta, \theta') = \log(\pi(\theta')/\pi(\theta))$$

is an intractable function of (θ, θ'). The acceptance probability in the s-algorithm for $\pi(\theta)$ is

$$\alpha(\theta, \theta') = \min(1, \exp(D(\theta, \theta'))).$$

Let $\hat{D}_{\theta,\theta'} = \hat{D}_{\theta,\theta'}(W)$ be an estimator for $D(\theta, \theta')$ computed from a collection of m random variables $W = (W_1, W_2, ..., W_m)$. Estimator $\hat{D}_{\theta,\theta'}$ has cdf $G_m(\cdot; \theta, \theta')$ and density $g_m(\cdot; \theta, \theta')$. Let

$$\sigma^2 = \lim_{m\to\infty} \mathrm{var}(\sqrt{m}\hat{D}_{\theta,\theta'}(W)),$$

so that the estimaor variance is asymptotically σ^2/m. For example, if the W_i are iid, and $\hat{D}_{\theta,\theta'}(W) = \overline{W}$, then $\sigma^2 = \mathrm{var}(W_1)$.

We begin with a 'naive' incorrect algorithm which is not itself an r-algorithm. We refer to an algorithm as *naive* when an estimate is plugged into the Metropolis Hastings acceptance probability, without correction. The 'MCWM' algorithm in Andrieu and Roberts 2009 [2] is an algorithm in this class. Although inexact, the algorithm may be useful. Beaumont 2003 [4] has shown that the approximation bias may be small, and the resulting (inexact) chain seems to have better mixing properties than the corresponding (exact) pseudo-marginal MCMC.

Algorithm 3 (Naive algorithm). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta, \cdot)$ and an estimate $x_t \sim g_m(\cdot; \theta, \theta')$.
(2) With probability

$$\alpha_{\mathrm{N}}(\theta, \theta') = \min\left\{1, \mathrm{e}^{x_t}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

The naive algorithm does not in general target π, and may not even have an equilibrium distribution.

The penalty method of Ceperley and Dewing 1999 [6] corrects the acceptance probability in the naive algorithm. A variant of this algorithm is used by these and other authors to simulate Gibbs distributions of interest in physical chemistry. Following Ceperley & Dewing 1999 [6]), we suppose a normal estimator $\hat{D}_{\theta,\theta'} \sim N(D(\theta, \theta'), \sigma^2(\theta, \theta')/m)$ is available. We assume $\sigma(\theta, \theta') = \sigma(\theta', \theta)$ and write $\sigma = \sigma(\theta, \theta')$ below. Ceperley & Dewing 1999 [6] show that the following algorithm, which they call the 'penalty method', targets $\pi(\theta)$ exactly.

Algorithm 4 (Penalty method (Ceperley & Dewing 1999 [6])). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta, \cdot)$ and an estimate $y_t \sim N(D(\theta, \theta'), \sigma^2/m)$.
(2) With probability

$$\alpha_P(\theta, \theta') = \min\left\{1, e^{y_t - \sigma^2/2m}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

Ceperley and Dewing 1999 [6] show that detailed balance is satisfied by carrying out the integrals over y_t needed to verify Eqn 2.4. We now show that the penalty method is an r-algorithm. Let X_t have a normal density, $\xi(x; \theta, \theta') = N(x; 0, \sigma^2 m)$ and take for f the involution $f(x) = \sigma^2 - x$. It follows that

$$\frac{\xi(f(x_t); \theta', \theta)}{\xi(x_t; \theta', \theta)} |f'(x_t)| = \mathrm{e}^{x_t - \sigma^2/2m}$$

and hence the acceptance probability in the r-algorithm at step t is

$$\alpha_\xi(\theta, \theta'; x_t) \min\left\{1, \mathrm{e}^{D(\theta,\theta') + x_t - \sigma^2/2m}\right\}.$$

Since $D(\theta, \theta') + X_t$ (in the r-algorithm) and Y_t (in the penalty method in Alg 5) have the same distribution, the algorithms are equivalent and the results in Sec 2.2 prove that the penalty method targets π.

Ceperley and Dewing 1999 [6] give other more general but approximate algorithms, the simplest of which is the Penalty Estimate method. We analyze this alongside the naive algorithm in Section 4. Suppose we have a unbiased normal estimator for D, but do not know its variance, and so try replacing σ^2 with the sample variance s^2.

Algorithm 5 (Penalty Estimate method (Ceperley & Dewing 1999 [6])). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta, \cdot)$, an estimate $y_t \sim N(D(\theta, \theta'), \sigma^2/m)$ and an independent variance estimate s^2 with

$$(m-1)s_t^2/\sigma^2 \sim \chi^2(m-1).$$

(2) With probability

$$\alpha_{\hat{P}}(\theta, \theta') = \min\left\{1, e^{y_t - s_t^2/2m}\right\}$$

set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

This algorithm is inexact. It is not an r-algorithm.

Ball et al. 2003 [3] give a 'universal rule' which applies for symmetric error distributions with compact support and does not require that the error variance σ^2 be known. In the case where the error distribution is normal the algorithm can target only a very good approximation to π. We can show that the universal rule for a normal error distribution is an r-algorithm, with two randomizations, $\xi_1(u_1; \theta, \theta') = N(u_1; 0, \sigma^2/m)$ and $\xi_2(u_2; \theta, \theta') = N(u_2; \sigma^2/m, \sigma^2/m)$ and the same involution, $f(u_i) = \sigma^2/m - u_i, i = 1, 2$, for each. In this bivariate randomization the unknown variance appears in a factor multiplying the whole acceptance probability. The acceptance probability may be simulated by a form of rejection. Promising though it is, we do not discuss the universal algorithm here, as there is no exact algorithm for the normal case and we make no further use of the connection.

The final algorithm we discuss is the *single variable exchange* algorithm of Murray & MacKay 2006 [15]. We present this in its original Bayesian setting, with data $d = (d_1, ..., d_n)$, prior $p(\theta)$ and Likelihood $L(\theta, d) = c(\theta)\tilde{L}(\theta, d)$. The target is now the posterior distribution,

$$\pi(\theta|d)) \propto p(\theta)L(\theta, d).$$

The likelihood has an intractable normalizing constant $c(\theta)$. Murray & MacKay 2006 [15] arrange things so that this cancels in the Hastings ratio, developing an idea due to Møller et al. 2004 [13]. However, while Møller et al. 2004 [13] augment the MCMC state with an auxiliary variable, the auxiliary variable in the exchange algorithm is associated with a single update, as in the r-algorithm.

Algorithm 6 (Single Variable Exchange). At state $\Theta_t = \theta$, simulate Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta, \cdot)$ and $x_t \sim L(\theta', \cdot)$.
(2) With probability

$$\alpha_E(\theta, \theta') = \min \left\{ 1, \frac{p(\theta')L(\theta', d)}{p(\theta)L(\theta, d)} \frac{L(\theta, x_t)}{L(\theta', x_t)} \right\}$$

set $\Theta_{t+1} = \theta'$ and otherwise set $\Theta_{t+1} = \theta$.

Factors of $c(\theta)$ and $c(\theta')$ cancel. This is an r-algorithm with $\xi(x; \theta, \theta') = L(\theta', x)$ and the identity involution $f(x) = x$. This is the first useful instance we have given for r-MCMC (ie, excluding the example in Section 2.4) in which ξ actually depends on θ'.

How do these identifications of existing algorithms as r-algorithms help us? Both Ceperley & Dewing 1999 [6] and Ball et al. 2003 [3] treat detailed balance as an integral equation, integrated over the random variation introduced by the estimator, $\hat{D}$. The acceptance probability is obtained as a solution of this integral equation. In fact the 'very detailed' balance relation Eqn 2.8 shows that the functions under the integrals are equal. We expect that this will help with the development of new algorithms.

4. Separation times and approximate-target MCMC

In this section we give a coupling strategy which motivates the use of the naive algorithm in some cases. We present and analyze a coupling algorithm to show that, on average, the naive algorithm gives *exactly the same* MCMC samples as the exact penalty method, out to $O(m)$ steps of the naive chain, where m is the sample size used in D-estimation in the naive chain. The error in this algorithm is analyzed in Andrieu and Roberts 2009 [2] from a different perspective.

Recall that $\hat{D}_{\theta,\theta'}$ is an estimator for $D(\theta, \theta')$ with cdf $G_m(\cdot; \theta, \theta')$. We do not assume $\hat{D}_{\theta,\theta'}$ is unbiased or normal. We do assume it satisfies a CLT, so that

$$G_m(x) = \Phi\left(\frac{x - D}{\sigma/\sqrt{m}}\right) + O(m^{-1/2}), \tag{4.1}$$

with σ^2 the asymptotic variance of $\sqrt{m}\hat{D}_{\theta,\theta'}$. For example, if $\hat{D}_{\theta,\theta'}$ is computed from a realization of a geometrically ergodic Markov chain $W = \{W_i\}_{i=0}^{\infty}$ and

$$\hat{D}_{\theta,\theta'} = \frac{1}{m}\sum_{i=1}^{m} W_i, \tag{4.2}$$

then Eqn 4.1 holds, subject to mild additional conditions specified in Kontoyiannis and Meyn (2003) [9]. Results of this kind may be used with the delta method to get asymptotically normal ratio estimators. If Eqn 4.1 holds for $\hat{D}_{\theta,\theta'}$, then it can be coupled to a normal estimator.

$$\begin{aligned}
\hat{D}_{\theta,\theta'} &= D + \frac{\sigma}{\sqrt{m}}\frac{\hat{D}-D}{\sigma/\sqrt{m}} \\
&= D + \frac{\sigma}{\sqrt{m}}\Phi^{-1}(G_m(\hat{D}) + O(m^{-1/2})) \\
&= D + \frac{\sigma}{\sqrt{m}}\Phi^{-1}(G_m(\hat{D})) + O(1/m), \qquad (4.3)
\end{aligned}$$

where $\Phi^{-1}(G_m(\hat{D}))$ is a standard normal random variable.

We now give the coupling algorithm. In our example, we couple the naive and penalty method chains. Couplings of this kind may be applied to other pairs of algorithms. The algorithm simulates the penalty method and also an indicator variable $B_t \in \{0,1\}, t = 1,2,...$ marking the times at which the naive chain separates from the penalty-method chain.

Algorithm 7 (Coupling algorithm: penalty method and naive algorithm). At state $\Theta_t = \theta$, simulate B_t and Θ_{t+1} as follows:

(1) Simulate $\theta' \sim q(\theta,\cdot)$ and an estimate $x_t \sim g_m(\cdot;\theta,\theta')$, and set

$$y_t = D + \frac{\sigma}{\sqrt{m}}\Phi^{-1}(G_m(x_t)).$$

(2) Simulate $V_t \sim U(0,1)$. Let

$$\alpha_{\mathrm{P}}(\theta,\theta';y_t) = \min\left\{1, \mathrm{e}^{y_t - \sigma^2/2m}\right\}.$$

If $V_t \le \alpha_{\mathrm{P}}$ then set $\Theta_{t+1} = \theta'$, otherwise set $\Theta_{t+1} = \theta$.

(3) Let

$$\alpha_{\mathrm{N}}(\theta,\theta';x_t) = \min\left\{1, \mathrm{e}^{x_t}\right\}.$$

If

$$\min(\alpha_{\mathrm{N}},\alpha_{\mathrm{P}}) < V_t \le \max(\alpha_{\mathrm{N}},\alpha_{\mathrm{P}})$$

then set $B_t = 1$ and otherwise set $B_t = 0$.

The Θ_t chain targets π exactly, as it is a penalty method chain. The two chains (naive and penalty method) separate when V_t falls between the two acceptance probabilities, since then the chains make different accept/reject decisions. If this first separation time is larger than the run length, then the naive algorithm realizes the same samples as the penalty method, and the error from using an inexact chain is undetectable at the overall MCMC precision. We emphasize that we cannot usually implement the coupling algorithm as we cannot in general compute $\Phi^{-1}(G_m(\hat{D}_{\theta,\theta'}))$.

We now give a lower bound on the mean time to separation, assuming that the chains start in equilibrium. Let $T = \min\{t \geq 1; B_{t-1} = 1\}$ be the first passage time to separation. We assume $\Pr(T < \infty) = 1$. Next, condition the process on a separation at the first trial, $B_0 = 1$, set $T_0 = 0$ and let

$$T_i = \min\{t > T_{i-1}; B_t = 1, i = 1, 2, 3, ...|B_0 = 1\}$$

be the sequence of separation return times. The intervals $T_i - T_{i-1}$ are the random intervals between separation events. The B_t process is not in general a Markov chain, and the intervals are not iid. However, conditional on $B_0 = 1$, the B_t process is a stationary discrete process, with interval-stationary separation return times. Let $\rho = \mathbb{E}(T_1)$ be the mean separation return-time. By Kac's Recurrence Theorem for a stationary discrete process, $\rho = 1/\Pr(B_0 = 1)$. Let

$$\mathbb{E}|\alpha_P - \alpha_N| = \int_{E^2 \times R} |\alpha_P(\theta, \theta'; \delta) - \alpha_N(\theta, \theta'; \delta)|\, g_m(\delta) d\delta Q(\theta, d\theta')\pi(d\theta).$$

The mean separation return-time is

$$\rho = \frac{1}{\mathbb{E}|\alpha_P - \alpha_N|}. \tag{4.4}$$

The separation time from the initialization $\Theta_0 = \theta_0$ is $\tau(\theta_0) = \mathbb{E}(T|\Theta_0 = \theta_0)$. Let $\tau = \int_E \tau(\theta)\pi(d\theta)$ give the mean separation time starting in equilibrium. This time is the expected first passage time to a separation event $B_t = 1$ from an equilibrium start for Θ_0. The return time of an interval-stationary process bounds the first passage time by $2\tau \geq \rho$, as the return time around a fixed time is length-biased, and the fixed time is uniformly distributed in an interval between separation events. It follows that the mean separation time τ grows at least linearly with increasing m, since $|\alpha_P - \alpha_N| = O(1/m)$ in Eqn 4.4. We have not bounded the separation time $\tau(\theta_0)$ from any particular start state. However we assume $\tau(\theta_0) \simeq \tau$ as the the event $\alpha_P = \alpha_N = 1$ occurs more frequently during convergence, and the chains cannot separate on these events.

If we couple the naive algorithm to the standard algorithm, s-MCMC, with acceptance probability α, we find $\mathbb{E}|\alpha - \alpha_N| = O(1/\sqrt{m})$. The naive algorithm is therefore 'closer' to the exact penalty method, in the 'distance' $\mathbb{E}|\alpha_P - \alpha_{\mathrm{N}}|$ than it is to the exact standard algorithm. We improve the naive approximation to the target by improving the approximation $G_m(x) \simeq \Phi\left(\frac{x-D}{\sigma/\sqrt{m}}\right)$.

The Penalty Estimate method (Algorithm 5) has separation times of $O(m^{3/2})$ at the price of stronger conditions on the distribution of the estimator $\hat{D}_{\theta,\theta'}$. The estimator $x_t \sim N(D(\theta,\theta'), \sigma^2/m)$ in the Penalty Estimate method is exactly normal, but we do not know σ^2, and proceed as in Algorithm 5. The estimator is normal, so it can be coupled to the (exact) Penalty Method using the trivial coupling $y_t = x_t$. Now since

$$\frac{(m-1)s^2/\sigma^2 - (m-1)}{\sqrt{2(m-1)}} \overset{D}{\to} N(0,1),$$

the two acceptance probabilities $\alpha_{\hat{P}}$ and α_P differ by terms of $O(m^{3/2})$. We give an example with this behavior in Section 5.

Finally we note that if the chains are independence samplers, the naive chain and the penalty method chain may separate and then coalesce. This kind of coupling is used in the 'perfect simulation' algorithm of Murdoch et al. 1998 [14]. The two chains are offered the same candidate at each step, even after separation, and they coalesce when they both accept. If the time to coalesce is much smaller than the time to separate, the naive algorithm may give a very good approximation indeed.

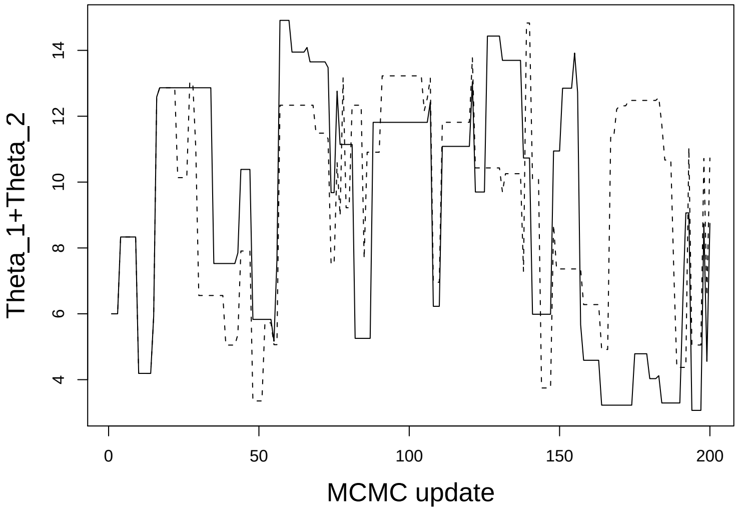
5. Examples

In order to show the linear dependence of the separation time τ on m, we give a very simple example for which we can compute $\Phi^{-1}(G_m(x))$. Let π be an equal mixture of bivariate normals, with $\theta = (\theta_1, \theta_2)$ and

$$\pi(\theta) = \frac{1}{2}\, MVN(\theta; \mu_1, \Sigma_1) + \frac{1}{2} MVN(\theta; \mu_2, \Sigma_2),$$

$\mu_1 = (3,3)^T$, $\mu_2 = (6,6)^T$, $[\Sigma_a]_{i,i} = 1$ for $a, i = 1, 2$, and $[\Sigma_1]_{1,2} = 1/2$ and $[\Sigma_2]_{1,2} = -0.5$. If $D = \log(\pi(\theta')/\pi(\theta))$ and for $i = 1, 2, ..., m$, $W_i \sim \mathrm{Exp}(1)$ then we use $\hat{D} = D - 1 + m/\sum_i W_i$ to estimate D. In this example $G_m(x) = F((x+1)/m; m, 1)$ where $F(\cdot; m, 1)$ is the cdf of an inverse Gamma variable with shape M and rate 1, so that $\mathbb{E}(\hat{D}) = D + 1/(M-1)$ and $\mathrm{var}(\hat{D}) = m^2/(m-1)^2(m-2)$. In this example, $\hat{D}$ is a biased, non-normal estimator for D.

In Fig 1 we plot realizations of coupled Penalty Method and naive



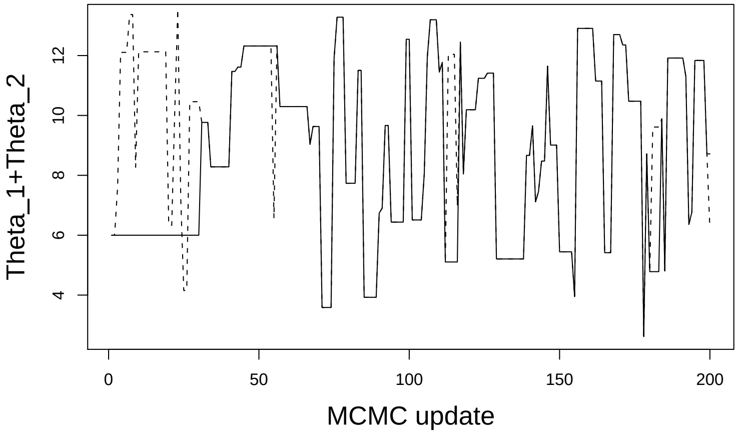


Figure 1. Simulations of the MCMC coupling-separation algorithm $\Theta_1 + \Theta_2$ in the Penalty Method (solid lines) and Naive Algorithm (dashed lines), (Top) with Random-Walk Metropolis updates and (bottom) with Independence-Sampler updates. Target density is a mixture of bivariate normals, D-estimator using $m = 8$ samples at each update.

MCMC samplers using random walk Metropolis, and an Independence sampler. The means of the mixture components are on $\theta_1 = \theta_2$ so we plot $\Theta_1 + \Theta_2$ against MCMC update to show the mixing across components. For illustration, we compute our estimator $\hat{D}$ from just eight independent W_i's (we set $m = 8$). This ensures that the mean separation times are short, approximately 72 updates for the random walk update and 32 for the independence sampler. This is convenient for graphical display. In this simulation the random walk proposal (top) for the Penalty Method and naive samplers separate after 25 updates. The walks shadow one another as they use the same uniform numbers to generate proposals. The Independence samplers separate after just one update, but then coalesce, and go on to branch and coalesce many times. In an Independence sampler run of length 10000 updates with $m = 8$, 90% of the samples returned by the naive sampler are exactly equal to those returned by the Penalty Method.

In Fig 2 we show the slight over-dispersion of the naive samplers compared to the target distribution. This is visible because we used a very small sample size for the D-estimator $\hat{D}$ of $m = 8$ iid samples. In this example, the Penalty method samplers are exact, even with $m = 8$, and follow the target density to within error.

The upper graph in Fig 3 demonstrates the linear dependence of the separation return times ρ and separation times τ on the number of samples m used to form $\hat{D}$. Two estimates of ρ are computed. The first is from Eqn 4.4,

$$\hat{\rho}_1 = \frac{1}{K^{-1}\sum_{t=1}^{K} |\alpha_P(\theta_t, \theta_t'; y_t) - \alpha_N(\theta_t, \theta_t'; x_t)|}.$$

For the second, $\hat{\rho}_2 = S^{-1}\sum_{i=1}^{S}(T_i - T_{i-1})$ estimates ρ. The $\hat{\rho}_1$ estimator has lower variance than the $\hat{\rho}_2$ estimator. The τ-estimator is the mean of 1000 realizations of T. The estimates are computed for Random-Walk Metropolis and Independence Samplers. The Naive Independence sampler separates more rapidly than the Naive Random-Walk sampler, on average, but can coalesce, so it generates estimates of similar bias in this example. The separation return times ρ and separation times τ are approximately equal. The separation mark process B_t is not in general Markov so $\mathbb{E}(T|B_0 = 1) \neq \mathbb{E}(T)$ in general. However in this example the separation return times $T_i - T_{i-1}$ are not easily distinguished from iid geometric random numbers.

In order to demonstrate the $O(m^{3/2})$ behavior of separation times between coupled Penalty Method and Penalty Estimate Method chains we take $W_i \sim N(0,1)$ $\hat{D} = D + \sum_i W_i/m$ an use $s^2(D_1, ..., D_m)$ to

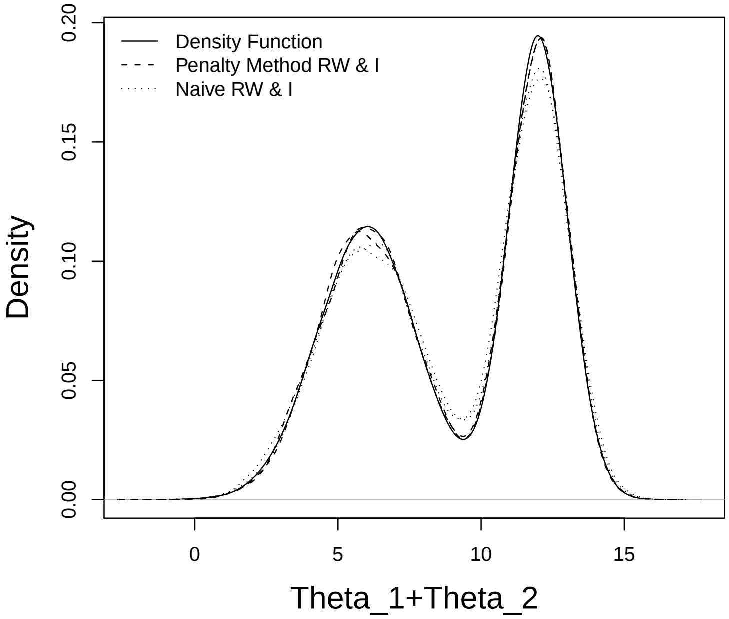


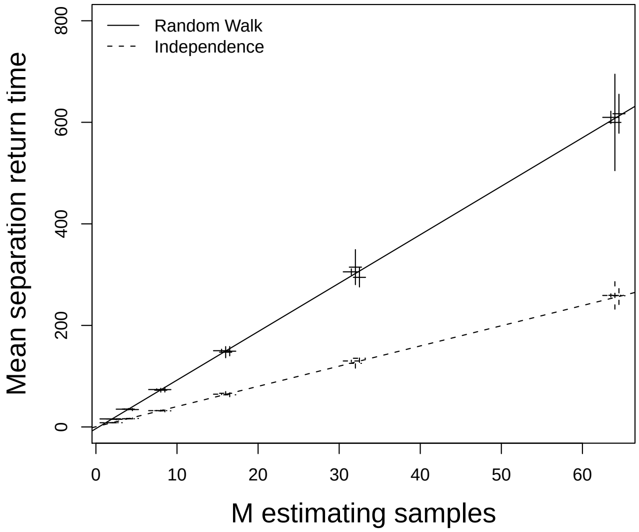
FIGURE 2. Density for $\Theta_1 + \Theta_2$ in the target mixture of bivariate normals (solid curve) plotted with densities estimated using the Penalty Method (dashed curves for Independence and Random Walk updates) and Naive Algorithms (dotted curves).

estimate σ^2. The estimate is unbiased and normal, but the variance is unknown. The separation time function $\tau = \tau(m)$ displayed in the lower graph in Fig 3 grows very rapidly with m.

6. Discussion

We have presented two new algorithms. The first is an 'exact' MCMC algorithm for a randomized acceptance probability, in which the randomization is not part of the MCMC state. As this algorithm generalizes the simplest forms of of the penalty method, universal algorithm and single variable exchange algorithms, it may have applications in the simulation of doubly intractable target distributions.

Randomized MCMC is complementary to pseudo-marginal MCMC (Lin et al. (2000) [11], Beaumont (2003) [4], Andrieu et al (2009) [2]) and Andrieu et al (2010) [1]. In both algorithms an intractable target distribution is estimated using auxiliary random variables. However,



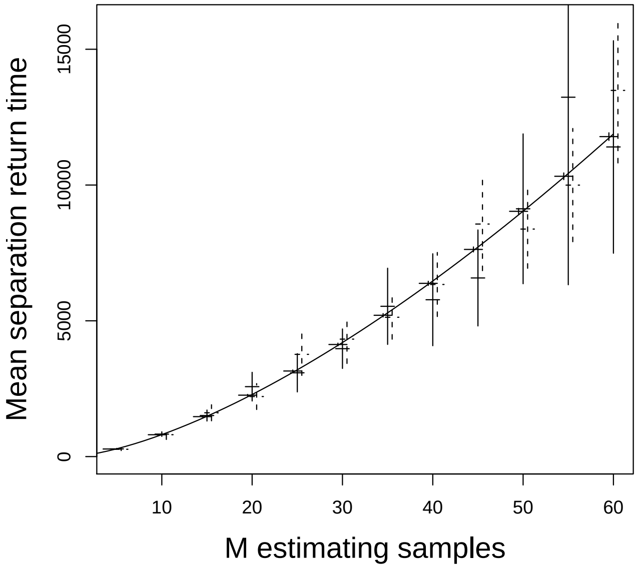


FIGURE 3. (Top) Estimated separation times ρ and τ between the exact Penalty Method and the approximate Naive Algorithm, as a function of estimator sample size m, for Random-Walk (solid lines) and Independence sampler (dashed lines) updates. Two estimates of ρ, $\hat{\rho}_1$ (left error bar in each group of three) and $\hat{\rho}_2$ (central error bar) and $\hat{\tau}$ (right error bar) are plotted for each sample and each m with a linear regression of the $\hat{\rho}_2$ estimates. (Bottom) Estimated separation times, as above, between Penalty Method and approximate Penalty Estimate chains regressed with $\hat{\rho}_1 = cm^{3/2}$.

in randomized MCMC algorithms the auxiliary variables extend the transition kernel, and it is the transition kernel which has the correct marginal at each update (ie, integrating Eqn 2.8), whereas in pseudo-marginal algorithms, the auxiliary variables extend the target distribution, and the target distribution has the correct marginal distribution. Randomized MCMC does not maintain auxiliary variables in the state, and this may be an advantage. Beaumont (2003)[4] notes that pseudo-marginal algorithms are prone to getting stuck when the target density is over-estimated. Simulation studies using very simple target densities and comparable exact pseudo-marginal and penalty method algorithm show the efficiency advantage of the Penalty Method increasing with increasing estimator variance. Against this, existing exact randomized algorithms impose much stronger conditions on the distribution of the estimator than pseudo-marginal algorithms.

In fact, the class of randomized Metropolis Hastings MCMC algorithms we have described are themselves special cases of a very large class of MCMC algorithms called Active Particle Algorithms, described in Lee et al. (2011) [10]. This class of algorithms includes the pseudo-marginal MCMC and randomized algorithms as special cases, as well as mixed algorithms in which both the target distribution and transition kernel in detailed balance are both marginals.

The second algorithm, the coupling-separation algorithm, is not directly useful for doubly intractable problems, as the simple form of the exact penalty method algorithm is not tractable. However, it shows that under some conditions the naive algorithm generates the exact same sequence of samples as an exact penalty method algorithm, out to $O(m)$ steps in the MCMC, where m is the sample size used to estimate the log of the Hastings ratio. This suggests strategies for improving naive simulation, using any method that tends to increase this separation time. The coupling-separation algorithm may help to suggest improvements in other settings involving an approximate likelihood in an MCMC algorithm. As an example, if we have a very large data set, and a log-likelihood given as sum over independent data, we may estimate the log-likelihood using a small sample of size m drawn with replacement from the data. The estimator is asymptotically normal in m.

The coupling-separation algorithm complements the perfect simulation algorithm of Propp and Wilson. In that algorithm coupled MCMC chains for the same target start in different states and coalesce. In the coupling-separation algorithm coupled MCMC chains for different targets start in the same state and branch.

Appendix A: Peskun ordering of s-chains and r-chains. We now prove the ordering given in Eqn 2.11. Let

$$A = \{x \in W : \xi(x;\theta,\theta') < h(\theta,\theta')\xi(f(x);\theta',\theta)|f'(x)|\},$$

so that

$$\alpha_\xi(\theta,\theta') = \int_A \xi(x;\theta,\theta')dx + \int_{W\setminus A} h(\theta,\theta')\xi(f(x);\theta',\theta)|f'(x)|dx. \tag{6.1}$$

Replacing $\xi(x;\theta,\theta')$ with $h(\theta,\theta')\xi(f(x);\theta',\theta)|f'(x)|$ for x in A,

$$\begin{aligned} \alpha_\xi(\theta,\theta') &\leq \int_W h(\theta,\theta')\xi(f(x);\theta',\theta)|f'(x)|dx \\ &= h(\theta,\theta'), \end{aligned}$$

while replacing $h(\theta,\theta')\xi(f(x);\theta',\theta)|f'(x)|$ with $\xi(x;\theta,\theta')$ for x in $W\setminus A$,

$$\begin{aligned} \alpha_\xi(\theta,\theta') &\leq \int_W \xi(x;\theta,\theta')dx \\ &= 1. \end{aligned}$$

It follows that $\alpha_\xi(\theta,\theta') \leq \min(1, h(\theta,\theta'))$.

Appendix B: irreducibility and minorization of r-chains. We now prove that Eqn 2.11 follows from Eqn 2.10. Let

$$\Xi(X) = \frac{\xi(f(X);\theta',\theta)}{\xi(X;\theta,\theta')}|f'(X)|$$

so that $\alpha_\xi(\theta,\theta';X) = \min(1, h(\theta,\theta')\Xi(X))$ and Ξ depends on θ and θ'. From the definition of $\alpha_\xi(\theta,\theta')$ in Eqn 2.3,

$$\begin{aligned} \alpha_\xi(\theta,\theta') &= \mathbb{E}\left(\alpha_\xi(\theta,\theta';X)\right) \\ &= \mathbb{E}\left(\alpha_\xi(\theta,\theta';X)|\Xi(X)\geq 1\right)\Pr(\Xi(X)\geq 1) \quad + \\ &\quad\ \mathbb{E}\left(\alpha_\xi(\theta,\theta';X)|\Xi(X)< 1\right)\Pr(\Xi(X)< 1) \\ &\geq \mathbb{E}\left(\alpha(\theta,\theta')|\Xi(X)\geq 1\right)\Pr(\Xi(X)\geq 1) \\ &= \alpha(\theta,\theta')\Pr(\Xi(X)\geq 1) \end{aligned} \tag{6.2}$$

where the inequality in the third equation holds because $\alpha_\xi(\theta,\theta';X) \geq \alpha(\theta,\theta')$ given $\Xi(X) \geq 1$.

Let

$$p_\xi(\theta,\theta') = \left\{ \begin{array}{cc} q(\theta,\theta')\alpha_\xi(\theta,\theta') & \theta \neq \theta' \\ 0 & \theta = \theta' \end{array} \right. \tag{6.3}$$

be the zeroed transition probability density in the r-algorithm. The condition in Eqn 2.10 gives $\Pr(\Xi(X) \geq 1) \geq \epsilon$ for some $\epsilon > 0$ not depending on θ, θ', and so $p_\xi(\theta, \theta') \geq \epsilon p(\theta, \theta')$ for the transition density, from Eqns 2.1, 6.2 and 6.3. Let

$$r_\xi(\theta) = 1 - \int_E p_\xi(\theta, \theta')\mu(d\theta')$$

give the probability to remain in the same state, in the r-algorithm. By the Peskun ordering, $r_\xi(\theta) \geq r(\theta)$, and so trivially, $r_\xi(\theta) \geq \epsilon r(\theta)$, and hence

$$\begin{aligned} P_\xi(\theta, B) &= \int_B p_\xi(\theta, \theta')\mu(d\theta') + r_\xi(\theta)\mathbb{I}_{\theta \in B} \\ &\geq \epsilon P(\theta, B) \end{aligned}$$

for each $\theta \in E$ and $B \in \mathcal{E}$. This is Eqn 2.11 and we are done.

If ξ does not depend on θ, θ' then condition Eqn 2.10 is not needed. In that case $0 < \Pr(\Xi(X) \geq 1) < 1$, since if

$$\xi(f(x))|f'(x)| < \xi(x)$$

for ξ-a.e. $x \in W$, then the two functions cannot both be probability densities, which is a contradiction. This implies Eqn 2.10 as there is no θ, θ'-dependence in $\Pr(\Xi(X) \geq 1)$. If there is θ' dependence, as in the single variable exchange algorithm in Section 3, then we need to exclude cases where $\Pr(\Xi(X) \geq 1) \to 0$ as θ' approaches a boundary of E, in order that Eqn 2.11 hold for ϵ independent of θ and C.

References

[1] C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *J. Roy. Statist. Soc Series B*, 72:269–342, 2010.

[2] Christophe Andrieu and Gareth O. Roberts. The pseudo-marginal approach for efficient Monte Carlo computations. *Ann. Statist.*, 27(2):697–725, 2009.

[3] Robin Ball, Thomas Fink, and Neill Bowler. Stochastic annealing. *Physical Review Letters*, 91(3):1–4, July 2003.

[4] M. A. Beaumont. Estimation of population growth or decline in genetically monitored populations. *Genetics*, 164:1139–1160, 2003.

[5] J. E. Besag, P. J. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statist. Sci.*, 10:3–66, 1995.

[6] D M Ceperley and M Dewing. The Penalty Method for Random Walks with Uncertain Energies. *Journal of Chemical Physics*, 110, 1999.

[7] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F on Radar and Signal Processing*, 140(2):107–113, 1993.

[8] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.

[9] I. Kontoyiannis and S. P. Meyn. Spectral theory and limit theorems for geometrically ergodic Markov processes. *The Annals of Applied Probability*, 13:304362, 2003.

[10] A. Lee, C. Andrieu, and A. Doucet. An active particle perspective of MCMC and its application to locally adaptive MCMC algorithms. 2011.

[11] L. Lin, K. F. Liu, and J. Sloan. A noisy Monte Carlo algorithm. *Phys. Rev. D*, 61:074505, Mar 2000.

[12] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092, 1953.

[13] Jesper Møller, A.N. Pettitt, K. K. Berthelsen, and R. W. Reeves. An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants. *Biometrika*, 93(2), June 2004.

[14] D. J. Murdoch and P. J. Green. Exact sampling from a continuous state space. *Scandinavian Journal of Statistics*, 25:483–502, 1998.

[15] Iain Murray and David J C MacKay. MCMC for doubly-intractable distributions. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2006.

[16] P.H. Peskun. Optimum Monte Carlo sampling using Markov chains. *Biometrika*, 60:607–612, 1973.

[17] James G. Propp and David B. Wilson. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *Journal of Algorithms*, 27:170–217, 1998.

[18] J.G. Propp and D.B. Wilson. Exact sampling with coupled Markov chains and applications to statistical mechanics. *Random Structures and Algorithms*, 9:223–252, 1996.

[19] L. Tierney. Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1762, 1994.

†: Department of Statistics, 1 South Parks Road, Oxford, OX1 3TG, UK, ∗: Department of Physics, University of Otago, NZ